



# Software Engineering Übung 5

Verträge, Aufwand- und Risikoschätzung

## 1 Informationen

### 1.1 Daten

- Ausgabe Di 10.11.2009
- Abgabe So 22.11.2009 bis 23:59 Uhr
- Besprechung am Di 01.12.2009

### 1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen **Ex[n]\_[NameA\_NameB\_NameC].pdf** abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA\_NameB\_NameC] die Nachnamen der Gruppenmitglieder sind. Die PDF Datei sollte ausserdem ebenfalls Ihre Namen und Matrikelnummern beinhalten.

Mailen sie Ihre Lösungen vor dem Abgabetermin an [wueest@ifi.uzh.ch](mailto:wueest@ifi.uzh.ch). Der Betreff der E-mail sollte mit **[SE EX HS09]** beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, mailen Sie bitte ein Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen sie das Archiv anhand der oben erwähnten Konventionen.

Die Übungen sollen in 3er Gruppen gelöst werden. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

## 2 Aufgabenstellung

Die Aufgaben beziehen sich auf die Fallstudie aus den Übungen 2 bis 4.

### 2.1 Vertragsformulierung (4 Punkte)

Gegeben ist die Schnittstelle der Java-Klasse `Tree`, welche benutzt wird, um verschiedene Kategorien von Aufführungen in einer baumartigen Struktur anzuordnen (eine Aufführung kann beispielsweise ein Film oder ein Theaterstück sein, ein Theaterstück wiederum kann eine Komödie, eine Tragödie, etc. sein).

Hinweis: Im Nachfolgenden bedeutet { ... }, dass die betreffende Methode eine Implementierung hat, diese aber hier nicht gezeigt ist.

Fügen Sie die fehlenden Verträge (jeweils Pre- und Postconditions sowie Klasseninvarianten) zur spezifizierten Schnittstelle hinzu. Für die bessere Verständlichkeit soll bei der Formulierung von Gleichheit in Ausdrücken von Verträgen das doppelte Gleichheitszeichen (==) verwendet werden. Verwenden Sie ansonsten die Notation, wie Sie diese aus der Vorlesung kennen (Folienkapitel 5.6). Sie müssen ausser den Verträgen keine weitere Implementierung vornehmen.

```
/**
 * Allows to link nodes in a tree-like structure. Nodes can be
 * attached to other nodes as children. Every child has exactly
 * one parent. An already attached node can not be attached again.
 *
 * A newly created tree consists of a root node. This node can
 * not be deleted.
 *
 * @inv
 *
 */
public class Tree {

    /**
     * The root node of the tree.
     */
    private Node root;

    /**
     * This vector contains all children of the tree.
     */
    private Vector<Node> children;

    /**
     * Checks whether this tree contains any elements or not.
     *
     * @pre
     * @post
     *
     * @return true if the tree is empty
     */
    public boolean isEmpty() {...}

    /**
     * Returns the size of the tree.
     *
     * @pre
     * @post
     *
     * @return the number of nodes in the tree
     */
    public int getSize() {...}
```

```

/**
 * Attaches a given node to the given parent.
 *
 * @throws NullPointerException
 *         if node is null or if parent is null
 *
 * @pre
 * @post
 *
 * @param node
 *         the node which gets attached
 * @param parent
 *         the parent to which the node gets attached
 */
public void attach( Node node, Node parent ) {...}

/**
 * Removes a node from the tree.
 *
 * @pre
 * @post
 *
 * @param node
 *         the node to remove from the tree
 */
public void remove( Node node ) {...}

/**
 * Checks whether a given node exists in the tree or not
 *
 * @pre
 * @post
 *
 * @param node
 *         the node to search for
 *
 * @return true if the node is found in the tree
 *         false otherwise
 */
public boolean exists( Node node ) {...}
}

```

**Sitzplatz-Kauf**

Durch Ausfüllen des Formulars kaufen Sie die gewählten Sitzplätze.

Name:

Adresse:

Email:

Kreditkarten-Nr.:  valid thru:

**Abbildung 1:** Oberflächenprototyp für den Online-Kauf von Sitzplätzen

## 2.2 Aufwandschätzung, Function Points (6 Punkte)

Gegeben sei der Oberflächenprototyp für den Online-Kauf von ausgewählten Sitzplätzen (siehe Abb. 1).

Die folgenden zusätzlichen Informationen zu dieser Eingabemaske sind gegeben:

- Alle Felder sind zu Beginn leer.
- Änderungen werden über die Schaltfläche *Bestätigen* gespeichert, und *Sitzplatz-Kauf* wird verlassen. Die Kreditkartennummer wird überprüft und die Zustände der Sitzplätze ändern sich.
- Mit der Schaltfläche *Abbrechen* wird *Sitzplatz-Kauf* verlassen. Änderungen, die nicht gespeichert wurden, gehen dabei verloren.

a) Ermitteln Sie die Adjusted Function Points gemäss der Vorgehensweise aus der Vorlesung. Beschreiben Sie dabei Ihre Vorgehensweise und dokumentieren Sie die Annahmen, welche Sie treffen. Die Berechnung der Function Points für den Oberflächenprototyp betrifft nur Dateneingaben und Anfragen, nicht die Datenausgaben. Nehmen Sie für die Anfragen, für die externen Schnittstellen und die internen Datenbestände jeweils, falls vorhanden, einen einfachen Schwierigkeitsgrad an. Für die Gewichtung der Eingaben können die Werte der Tabelle auf Folie 22, Kapitel 16 aus der Vorlesung analog verwendet werden. Bei der Berechnung des Gesamteinflussfaktors werden die Effizienz der Benutzerschnittstelle, die Wiederverwendbarkeit, die einfache Benutzbarkeit und die Erweiterbarkeit als sehr hoch bewertet. Komplexe Verarbeitungen und Installationen an mehreren Orten haben keinen Einfluss. Für die restlichen Faktoren wird ein durchschnittlicher Einfluss angenommen. Falls Sie noch weitere Annahmen treffen, dokumentieren und begründen Sie diese.

b) Was brauchen Sie neben den Adjusted Function Points noch, um eine Schätzung des Aufwandes für den Teil unter a) zu machen? Begründen Sie die Aussagen kurz.

### 2.3 Aufwandschätzung, COCOMO2 (6 Punkte)

Die Firma IT4U will sich um den Auftrag, das System für *Ambidramas* zu entwickeln, bewerben. Bevor man ein Angebot macht, wird intern eine Aufwandschätzung vorgenommen. Verschiedene Möglichkeiten werden verglichen und ihre Auswirkungen auf den Aufwand werden berechnet. Als algorithmisches Schätzverfahren wird COCOMO2 verwendet.

a) Das Projektteam besteht aus Mitarbeitern, welche bereits gemeinsame Projekte bearbeitet haben. Projekte, die dem *Ambidramas* Projekt ähnlich sind, gab es allerdings noch keine. Für den Posten des Projektleiters gibt es zwei Bewerbungen. Die erste Bewerbung ist von einem Mitarbeiter, der schon lange bei IT4U arbeitet. Er macht sich grosse Hoffnungen, den Job zu erhalten, denn er möchte Erfahrungen im Leiten von Projekten sammeln. Falls er den Posten nicht bekommt, wird er als Teammitglied beim Projekt dabei sein.

Die andere Bewerbung kommt von einer Mitarbeiterin von OnlineSystemsIT, einer Firma welche sich zum Teil auf Software im Online-Ticketing-Bereich spezialisiert hat. Sie hat Erfahrung in der Entwicklung von ähnlichen Theater-Systemen.

Welche Skalierungsfaktoren und Kostenfaktoren sind von der Wahl des Projektleiters betroffen? Diskutieren Sie die Auswirkungen beider Wahlmöglichkeiten auf diese Faktoren.

b) Da bisher in der Praxis schon viele Online-Ticketing-Systeme mit dem J2EE Framework implementiert wurden, soll, zu Zwecken besserer Wiederverwendbarkeit vorhandener Software und Kompatibilität zu vorhandenen Schnittstellen, die Software mit diesem Framework entwickelt werden. Es gibt einen Programmierer in der Firma, dem sehr gute Fähigkeiten im Programmieren mit Java und J2EE zugesprochen werden. Es müssten aber noch weitere Mitarbeiter geschult werden, um das *Ambidramas* Projekt bewältigen zu können. Alternativ könnte die gesamte Programmierung an das indische Software-Unternehmen ITIndia outgesourct werden. ITIndia stehen viele mit J2EE erfahrene Programmierer zur Verfügung.

Welche Skalierungsfaktoren und Kostenfaktoren sind von der Entscheidung, wo die Programmierung stattfinden soll, betroffen? Diskutieren Sie die Auswirkungen beider Möglichkeiten auf diese Faktoren.

c) Sie müssen den Skalierungsfaktor *Flexibilität* beurteilen. Nennen Sie zwei Eigenschaften des Projekts, welche dabei eine Rolle spielen und beurteilen Sie diese anhand des gegebenen Projektes. Falls Sie dazu Annahmen treffen müssen, begründen Sie diese.

d) Um wieviel Prozent müsste der Berechnungsfaktor KSLOC (Kilo Source Lines Of Code) verringert werden können, damit sich Wiederverwendung, welche den Wert des Kostenfaktors *Reuse Required* von Nominal auf Very High ändert, lohnt? Die restlichen Kostenfaktoren bleiben gleich und die Skalierungsfaktoren werden alle mit Nominal bewertet. Zur Lösung soll auch der Rechnungsweg dargestellt werden.

e) Was ist Voraussetzung, damit die Berechnung mit COCOMO2 zuverlässige Werte liefert? Begründen Sie Ihre Aussage kurz.

## **2.4 Risikoschätzung (4 Punkte)**

Im Skript Software Engineering zur Vorlesung finden Sie im Kapitel 15 die zehn häufigsten Risiken bei Software-Projekten.

**a)** Finden Sie 4 weitere Risiken für Ihr Softwareprojekt aus Übung 2 (keines der Risiken aus dem Skript).

**b)** Nehmen Sie eine Risikobewertung für die vier weiteren gefundenen Risiken vor, indem Sie das Risiko, sowie die Schadenshöhe je auf einer Skala zwischen 1 - 10 bewerten.

**c)** Stellen Sie passende Massnahmen zusammen, um die Risiken zu mindern oder zu eliminieren.